DOCUMENT RESUME

ED 413 900 IR 056 730

AUTHOR	Vorndran, Gerald
TITLE	A Search Engine Features Comparison.
PUB DATE	1997-08-00
NOTE	48p.; Master's Research Paper, Kent State University.
PUB TYPE	Dissertations/Theses (040) -- Reports - Evaluative (142)
EDRS PRICE	MF01/PC02 Plus Postage.
DESCRIPTORS	Access to Information; Computer Interfaces; Databases; *Online Searching; *Online Systems; Online Vendors; Reference Services; Relevance (Information Retrieval); Search Strategies; *User Needs (Information); *World Wide Web
IDENTIFIERS	*DIALOG; *Search Engines

ABSTRACT

Until recently, the World Wide Web (WWW) public access search engines have not included many of the advanced commands, options, and features commonly available with the for-profit online database user interfaces, such as DIALOG. This study evaluates the features and characteristics common to both types of search interfaces, examines the Web search interfaces to define lingering deficiencies as compared to the online interfaces, and presents suggestions for improvement to those areas of the Web interfaces found lacking. The most advanced interface features of the AltaVista, Excite, HotBot, and Infoseek Web search interfaces were compared to the DIALOG interface features. The Web search interfaces, as a whole, still trail the DIALOG search interface in terms of the quality, quantity, depth (robustness), and usability of the search system. Appendices include background information, search parameters, and output for the Web engines, and for DIALOG. (Contains 46 references.) (Author/SWC)

****
* Reproductions supplied by EDRS are the best that can be made *
* from the original document. *
****

IR

ED 413 900

A Search Engine Features Comparison

A Master's research Paper submitted to the Kent State University School of Library Science and Information Science in partial fulfillment of the requirements for the degree Master of Library Science

by

Gerald Vorndran

August, 1997

PERMISSION TO REPRODUCE THIS MATERIAL HAS BEEN GRANTED BY

R. Du Mont

TO THE EDUCATIONAL RESOURCES INFORMATION CENTER (ERIC)

U.S. DEPARTMENT OF EDUCATION
Office of Educational Research and Improvement
EDUCATIONAL RESOURCES INFORMATION CENTER (ERIC)

☐ This document has been reproduced as received from the person or organization originating it.

☐ Minor changes have been made to improve reproduction quality.

• Points of view or opinions stated in this document do not necessarily represent official OERI position or policy.

IR056730

BEST COPY AVAILABLE

Master's Research Paper by

Gerald K. Vorndran

B.A., Kent State University, 1989

M.L.S., Kent State University, 1997

Approved by

Advisor _____ Date 8/4/97

A Search Engine Features Comparison

Gerald Vorndran

Abstract

The World Wide Web (WWW) public access Search Engines have until recently not included many of the advanced commands, options, and features commonly available with the for-profit Online database user interfaces. To date, there has not been a comprehensive comparative analysis of the popular Web Search Engine search interface features to the established (fee) Online database vendor search interface features. This study presents, discusses, and evaluates those features and characteristics common to both types of search interfaces, examines the Web search interfaces to define lingering deficiencies as compared to the Online interfaces, and presents suggestions for improvement to those areas of the Web interfaces found lacking. The most advanced interface features of the AltaVista, Excite, HotBot, and Infoseek Web search interfaces were compared to the DIALOG interface features. It was found that the Web search interfaces, as a whole, still trail the DIALOG search interface in terms of the quality, quantity, depth (robustness), and usability of the search system.

Table of Contents

I. Introduction 1
- Scope 1
- Objectives 1
- Definitions 2
- Limitations 5

II. Literature Review 6

III. Methodology 8

IV. Research Findings 10
- Database Characteristics 10
- Search Interface Characteristics 13
- Commands 20
- Output 25
- Sample Searches 28
- Analysis of Search Engine Features 31

V. Summary and Conclusions 33

VI. Appendixes 34

VII. Bibliography 40

I Introduction

The World Wide Web (WWW) has experienced a proliferation of new public use Search Engines since its inception. New Search Tools have been appearing regularly and established Engines have been updated with new and more useful features on a regular basis due to increased competition from competing Web Searching Tools. Thusfar, however, those Search Engines freely available on the Web have not included, or approached the relative power of, the advanced features commonly available on the for-profit interfaces in the areas of quality, quantity, depth, or usability.

Scope

Need for Conducting this Study

Numerous studies comparing and contrasting these search aids have appeared in both the professional literature and popular trade journals. While there is a continuing need for such studies as new Search Engines become available for public use and old ones are upgraded with the newest features, there is currently a saturation of literature comparing the present free public use Web Engines. What has not been accomplished to date is a comprehensive comparative study of the popular Web Search Engine search interface features to the established (fee) Online database vendor search interface features.

Although there is not a direct correlation between Web and Online search interfaces, there is enough overlap between the numerous features of the two to warrant a comparative study. For example, although we would not search for Web URLs through an Online interface, both systems employ Boolean operators as a searching tool or technique. How the latter work via the interface, and to what degree they are available, can be compared and contrasted however.

Objectives

Questions to be Answered by, and Purpose of this Study

The goal of this study is to present, discuss, and evaluate those features, commands, options, search language syntax, and characteristics common to both types of search interfaces, examine the Web search interfaces to define deficiencies as compared to the Online interfaces, and present suggestions for the improvement to those areas of the Web interfaces found lacking. It is assumed that the Web Search Engines are not yet comparable to their Online counterparts in the variety of options available or depth of available search features.

Definitions

The words ENGINE, SEARCH ENGINE, SEARCH INTERFACE, and SEARCH TOOL have been used interchangeably.

The words ONLINE INDUSTRY, ONLINE VENDORS, DATABASE VENDOR(S)and VENDOR(S) have been used interchangeably.

The words RECORD and DOCUMENT have been used interchangeably.

Basic Index DIALOG Guide to Searching

In DIALOG, the index which typically contains all of the subject-related words. Fields that contain terms which describe the topic of the record.

Boolean Operator Electronic Computer Glossary

One of the Boolean logic operators such as AND, OR and NOT.

Boolean Search Electronic Computer Glossary

A search for specific data. It implies that any condition can be searched for using the Boolean operators AND, OR and NOT.

Contextual Search Electronic Computer Glossary

To search for records or documents based upon the text contained in any part of the file as opposed to searching on a pre-defined key field.

DIALOG Electronic Computer Glossary

An online information service that contains the world's largest collection of databases.

E-Mail Newton's Telecom Dictionary

A colloquial term for electronic mail.

A term which usually means Electronic Text Mail, as opposed to Electronic Voice Mail or Electronic Image Mail. Sometimes electronic mail is written as email. These days electronic mail is everything from simple messages flowing over a local area network from one cubicle to another, to messages flowing across the globe on an X.400 network. Such messages may be simple text messages containing only ASCII or they may be complex messages containing embedded voice messages, spreadsheets and images.

Engine Electronic Computer Glossary

Software that performs a primary and highly repetitive function such as a database engine, graphics engine or dictionary engine.

FAQ Newton's Telecom Dictionary

Either a Frequently Asked Question, or a list of frequently asked questions and their answers. Many Internet USENET news groups, and some non-USENET mailing lists, maintain FAQ lists (FAQs) so that participants won't spend lots of time answering the same set of questions.

Graphical User Interface Newton's Telecom Dictionary

GUI. A fancy name probably originated by Microsoft which lets users get into and out of programs and manipulate the commands in those programs by using a pointing device (often a mouse).

Hit Author Definition

Another name for a document retrieved as the result of a database search whether through a Web based engine or an Online Vendor.

HTML Electronic Computer Glossary

(HyperText Markup Language) A standard for defining hypertext links between documents. It is a subset of SGML (Standard Generalized Markup Language) and is used to establish links between documents on the World Wide Web.

HTML Newton's Telecom Dictionary

HyperText Markup Language. This is the authoring software language used on the Internet's World Wide Web. HTML is used for creating World Wide Web pages

Hypertext Newton's Telecom Dictionary

Also called hypermedia; software that allows users to explore and create their own paths through written, visual, and audio information. Capabilities include being able to jump from topic to topic at any time and follow cross-references easily. Hypertext is often used for Help files.

Interface Electronic Computer Glossary

The connection and interaction between hardware, software and the user.
Software, or programming, interfaces are the languages, codes and messages programs use to communicate with each other and to the hardware. Examples are the applications that run under the Mac, DOS and Windows operating systems as well as the SMTP e-mail and LU 6.2 communications protocols.
User interfaces are the keyboards, mice, commands and menus used for communication between you and the computer. Examples are the command lines in DOS and UNIX and the Mac, Windows and Motif graphical interfaces.

Internet Newton's Telecom Dictionary

Internet is a computer network which joins many government and university and private computers together over phone lines (mostly T-1s and T-3s). In 1995 the Government Accounting Office (GAO) said that Internet linked 59,000 networks, 2.2 million computers and 15 millions users in 92 countries.

Online Electronic Computer Glossary

Available for immediate use. If your data is on disk attached to your computer, the data is online. If it is on a disk in your desk drawer, it is off-line. If you use an online service, such as CompuServe or PRODIGY, you are online when you have made the connection via modem and logged on with your account number. When you log off, you are off-line.

Online Industry Electronic Computer Glossary

The collection of service organizations that provide dial-up access to databases, shopping, news, weather, sports, e-mail, etc.

Uniform Resource Locator Newton's Telecom Dictionary

URL. A standardized way of representing different documents, media and network services on the World Wide Web. An Internet term.

User Interface Electronic Computer Glossary

The combination of menus, screen design, keyboard commands, command language and help screens, which create the way a user interacts with a computer. Mice, touch screens, and other input hardware is also included.

Web Newton's Telecom Dictionary

An abbreviation for the Internet's World Wide Web. See WORLD WIDE WEB and INTERNET.

Web Browser Electronic Computer Glossary

A utility used to peruse documents on the Worldwide Web of the Internet.

World Wide Web Newton's Telecom Dictionary

Also called WEB or W3. The World Wide Web is the universe of accessible information available on many computers spread through the world and attached to that gigantic network called the Internet. The Web has a body of software, a set of protocols and a set of defined conventions for getting at the information on the Web. The Web uses hypertext and multimedia techniques to make the web easy for anyone to roam, browse and contribute to. The Web makes publishing information (i.e. making that information public) as easy as creating a "home page" and posting it on a server somewhere in the Internet.

Limitations

Due to the recent proliferation of studies which compare and contrast Web Search Engines to each other, their features, qualities, and the relevancy of their document retrieval, this study has not been conducted for the purpose of doing such a comparison. Nor has this study seeked to compare or contrast the document sources, the database quality, nor the indexing techniques of the databases themselves. It was not intended to evaluate the usefulness of a given database for a given search. It is simply meant as a comparative analysis of the options and features available to the user in querying the database. Furthermore, this study has not discussed the relative merits of Command Line searching versus Graphical User Interface (GUI) searching. It is those options available in accessing the database which are of prime focus, not necessarily the relative ease of use, or simplicity of, the interface features. Research was performed, and this study conducted, starting in June of 1996 and ending in March of 1997.

II Literature Review

As stated previously, numerous studies comparing and contrasting Web Search Engines have appeared in both the professional literature and popular trade journals. Surprisingly, there's also a substantial amount 'self published' work available exclusively on the Internet, some, to be expected, of better quality than others. Indeed, most of the literature appearing thusfar has appeared Online (unreviewed) and in the General Computer Trade Publications (semi-reviewed). These publications include Internet World, Mac World, Online, Online User, PC Computing, PC Magazine, and PC World among others. The professional literature, while slow to respond to evaluating the Web Tools initially, seems to now be responding to the void in the literature that had existed previously (Chu 1996, Ding 1996, and Srinivasan 1996).

The majority of the literature dealing with Web Search Engines falls into one of three categories, 'Feature tabulations', 'Engine rankings', and 'How to use' papers. Characteristic of this first type of Web Engine paper is a tabular comparison listing the features, options, and databases features of numerous Web Engines (Liu 1996, Overton 1996, Page 1996, Tillman 1996, and Westera 1996). Normally, a short description of each engine along with historical and statistical information and perhaps a simple description is provided for each engine being presented. Often, the relative merits or pros and cons is listed for each engine in the table. Usually there is no in-depth discussion of how each engine works, no sample searches performed, and no comparison or relative rating given. Countless such 'Feature tabulation' papers have proliferated throughout the Internet of late.

The second type of paper commonly written, the 'Engine ranking' type paper, will normally include some sort of in-depth comparison of the top engines, sample searches performed on those chosen engines, and tips or tricks to using said engines more efficiently (Chu 1996, Ding 1996, Morgan 1996, Venditto 1996, Westera 1996, Zorn 1996). In one way or another, they commonly go beyond the concise evaluations done in 'Feature tabulation' papers, although some overlap can, and does occur, between the two types.

The third type of Web Search Engine paper, the 'How to use' paper, will show the user how to exploit the engines for maximum proficiency. They will normally explain how an engine works, is indexed, and detail its most unique features (Conte Jr. 1996, Fleishman 1996, and Tillman 1996). They may or may not overlap in content area with the other two types of papers (Tillman 1996) but focus more on what the engine can do for the user and how it can help the user as a search tool.

Other, smaller bodies of literature, concern themselves with differing aspects of Online and Off-line search interfaces. Solock (1996) and James-Catalano (1996) discuss searching the Internet using Subject Catalogs, Subject Guides, and Annotated Directories. Duncan (1997) presents an analysis of automated search tools for off-line Web browsing and Carr (1996) introduces us to Intranet search software. Neither are applicable to this

study as they both discuss Commercial software which is not widely in use by the professional searcher. Notess (1996) explains the Internet 'Onesearch' search engines which allow the user to search all of the prominent search tools through a single interface, while Srinivasan (1996) and company propose a model to assist in the understanding of Web indexing. Finally, Feldman (1996) gives us a comparison of searching interfaces using DIALOG, TARGET, and DR-LINK software to rate advanced Natural Language database searching.

While there appears to be sufficient literature covering all aspects of the Web Search Engines and interfaces, what has not been accomplished to date is a comprehensive comparative study of the popular Web Search Engine search interface commands and features to the established Online database vendor interface commands and features. No literature at all, whether in the professional publications, trade journals, or Online sources, could be found which performs a study similar to the one proposed here.

III Methodology

General Methodology

This study has been conducted as a comparative analysis of various Search Engine characteristics. Each of the interface features or options below were evaluated by means of either direct database queries from an Online and a Web interface or through evaluation by the author.

The most advanced features and commands from each Web Engine are discussed in relation to a single Online Vendor's most advanced features and commands. As no single Web Engine contains all of the advanced features commonly available from the Online Vendor's more powerful engine, selected advanced features from each of the chosen Web Engines have been used in comparison. There is no direct Web Engine to Online Engine comparison performed for this reason. Only selected features of each are compared to form a conclusion as to the advanced state, or lack thereof, of the Web search interfaces.

Data was collected via inspection of the Engines Online help documentation and FAQ pages as well as the professional and non-professional literature which includes search engine comparisons appearing in the popular computer trade journals. Data were also collected via the direct submittal of sample queries to the Engines to determine exactly how they would parse a given search. DIALOG help documentation listed in the literature review was also consulted.

Selected Search Engines

Among the most popular and widely used, the AltaVista (http://www.altavista.digital.com), Excite (http://www.excite.com), HotBot (http://www.hotbot.com), and Infoseek (http://www.infoseek.com) Web based Search Engines are widely thought to be among the best available and have the most advanced features (Chu (1996), Overton (1996), and Zorn (1996)). Although others may be more widely used or more popular (Yahoo), they do not contain the quantity or quality of searching features, commands and search options as do these four. Also, it is of no consequence whether an engine evaluates documents in a subject directory fashion or not. It is the number of points of entry into the database and the advanced nature or quality of those features that is preeminent. An inverted index is not mandatory but the database had to have been searchable in some fashion.

Selected Online System

The DIALOG system search interface software was selected as the standard to compare the Web based features to. The DIALOG system has long been regarded as the industry leader in introducing new and innovative search tools to use in querying its databases. Although some vendors are catching up (notably Lexis-Nexis) in terms of advanced database search options and entry points, DIALOG remains a leader in this field and is

widely used by a diverse population which should benefit from this paper as more information seeking is done via the Web.

IV Research Findings

Database Characteristics

It is of foremost importance that we first discuss the two types of database search interfaces, how they are used, and what they are used for as well as the databases themselves, so as to better understand the interfaces. Obviously, we're discussing two media which are not directly comparable. For example, one would not search for a URL or Host Name using any DIALOG database. The Web Engines, though, do provide a searchable field to find URLs. There is no way to compare this feature on the Web Engines to any comparable DIALOG feature. We can only compare the most advanced Web database commands and features to those directly related DIALOG commands and features. But first we need to explore the databases themselves in order to more fully understand the interface features and commands.

Fee versus Free

Is there a fee for using a given service or are documents available for free? While it is normally true that you get what you pay for, that doesn't necessarily apply with Search Engines. Most engines either employ advertising to help pay for their service or are providing their engine free as a publicity tool for their product(s). In general, the Online Web databases can be searched for free using the publicly available search tools on the Web. The only major Search Engine that charges a fee is the Infoseek advanced user interface although the simple search feature is still free of charge. The Online vendors, of course, charge for the use of their services and one would expect them to have the most advanced search features available. Although some of the free Web interfaces may have started out as public service projects (at Universities), most are now affiliated with a for-profit company. We can therefore conclude that both the Web and Online vendors are in the Search Engine game for the same reasons and may eventually directly compete with each other.

Authority Control

Is there a form of authority control for Web documents? For the most part there is no form of authority control for Web documents available to be searched for by the Web Engines. The source and quality of documents is not known and not evaluated except for those documents available in subject tree format and included in the engines searchable indexes. Even these documents, though they may be reviewed, are not peer reviewed by professionals in a particular field. They may simply be categorized, rated, or classified by a company employee whose expertise may lie in another area. Documents from the Online Vendors may or may not come from peer-reviewed sources but one can always assume only highly 'relevant' documents in a given database will appear in that database. The vendors are not in the business of providing junk to their customers. On the Web, absolutely anyone can publish absolutely anything to be included as a Web document to be searched for by the Web Search Engines. While not a major concern in the evaluation of

search interfaces, it is still important to be aware of for this study in order to more fully understand how the interfaces are constructed and used.

Apples and Oranges

As discussed previously, the databases themselves are rather disparate in the information they contain. Maybe the analogy of 'apples and grapes' would be more accurate. While the Web indexes can be thought of as one huge database containing similar HTML documents (apple), the DIALOG user interface can access many individual databases containing significantly differing types of information (grapes). This was important to understand in distinguishing more precisely which features this paper would cover. Because not all commands or options are available with every database in DIALOG, we concentrated on only those basic commands readily available in all (or most) databases. The scope, therefore, of the databases we searched is not the same for each corresponding interface. There are no comparable DIALOG databases to search for Web documents. Where Web Search Engines will search nearly the same databases and can thus be seen as direct comparisons of the search software, the two systems we have looked at (Web Engines/DIALOG) have very little overlap in their database content.

User Interface

A mention needs to be made concerning the differences in the types of user interfaces compared. The Web based interfaces are all graphical interfaces (GUI) with no command option available. Fields and text boxes are selected by clicking with the mouse. The DIALOG system is a command driven interface which expects a certain syntax to be entered on the command line. There is a graphical equivalent known as Dialink which was not available for use with the Windows95 operating system at the time of this writing. Even so, it was still possible to compare the options and commands directly as they are widely used today.

Known Documents

Although we didn't perform sample searches to evaluate the quality of the information contained in the databases, again, we would still want to know something about the data itself in order to better comprehend how the interfaces are constructed. The Web Search Engines do give us some indication as to the number of documents indexed by their robots though the numbers are forever changing. Also, nobody knows exactly, and some would say even approximately, the number of Web documents or pages available to be cataloged and indexed. The Online vendors, conversely, know exactly how many documents are in their index and can guess as to the number of relevant documents left out (but they won't tell you). They know which journals or other relevant sources they do not have access to and therefore approximately how many documents are not included in their database. On the other hand, there is no way of knowing how many Web documents have been left out by the robots. Some servers deny access to robots while others have no links to them and thus no way for the robots to find the server in the first place. Again,

we were not attempting a qualitative analysis of the indexes but simply trying to understand the material to better understand the interfaces and their associated searching options or lack thereof.

Database Update Frequency

The DIALOG databases are updated on a regular, fixed basis. The Web Search Engine indexes claim to be updated on a regular basis but are not always. The Online documentation usually doesn't include this information and the user is left guessing as to how often the information is updated.

Full-text Searching

The full-text of the documents available through the four selected Web Interfaces and most DIALOG databases can be used to search the full-text of all documents available.

Word/Phrase Indexing

Options are available for both word and phrase searching in both the Web and DIALOG interfaces. However, not all Web interfaces support phrase searching at this time and this feature is still bundled in the advanced search modes and considered a feature to be used by expert searchers.

Vocabulary Control

Even though the <META> tag is readily available for use in any HTML document, it is still not widely used or supported. Thirty of the DIALOG databases do have online thesauri for the user to search on. The Web Search Engines have no such thesauri and the user is left to guess index words to employ in their search strategy.

Online Help

Both types of systems do offer some form of online help with the DIALOG help being more extensive. One thing that the Web based engines do well is that they normally provide for search examples as well as simple command definitions. These help to reinforce and solidify search strategies, especially for the novice user.

Search Interface Characteristics

A general comparison of the most advanced search options available on the Web and how they match up to the DIALOG features is discussed in this section. These are the general search characteristics and options which are the rules that the user must adhere to, must know about, and be concerned with, when parsing a query. We present issues concerning what the two types of engines can and can not do. These are not necessarily the actual commands used by the engines, but the rules which effect or govern efficient searching.

Default Search

Is there a default search for the interface? DIALOG never performs a default search as it is command driven though it expects some form of Boolean search to be entered. The Web interfaces offer a confusing array of default searches. Some default query boxes expect a full Boolean search to be entered. Others expect the user to enter only words which are then ANDed or ORed together. It is not always clear what type of search is to be performed. None of the Web Engines readily reveal the type of search that is expected by the Search Engine. The user should expect to be able to perform a full Boolean search on the default search page. Although the Web Engines provide for simple word searches as defaults, it is confusing for the novice to digest all of the available options. A uniform full Boolean default screen should be the norm for all engines as it provides the greatest functionality and would help if all Engines default search screens were standardized.

Boolean Operators

Which Boolean operators are available for use? The commonly available Boolean operators are the AND, OR, and NOT operators. DIALOG and most of the Web Engines allow for the use of these operators though the AND NOT operator may substitute for the NOT operator. Another substitution instituted by the Web Engines is the use of the plus (+) and minus (-) signs in place of the AND and NOT operators. Actually, to be more precise, the plus sign placed immediately in front of a term means that that term must appear somewhere in the document being searched for. This would be equivalent to simply entering a term on the command line in DIALOG (i.e. *ss auto*). Either the term is contained in the document or it is not. Some of the Web Engines (excite, Infoseek) use a fuzzy Boolean logic which allows for the retrieval of documents that do not contain an ANDed term. Hence, the introduction of the plus symbol. It forces the Engine to retrieve only documents containing that particular word. DIALOG automatically assumes that the user wants the word to appear somewhere in the document. The minus sign is simply a direct substitute for the NOT operator. Placing the minus sign in front of a word eliminates all documents containing that word from the retrieval listing. What the Web Engines which support these symbols are attempting to do is to combine the best of Boolean searching with a fuzzy logic (concept) search. Even though it may confuse some, this type of searching approach is appropriate for searching a database as large as the Web indexes have become. In such a large database (and with most Web pages now entering

extra words to generate more hits) it may be more efficient to use this type of fuzzy Boolean in order to retrieve a more manageable hit list. This is one of the areas that the Web based Engines may have a lead over the DIALOG engine.

Operator Order of Precedence

It is important to understand how an Engine parses a query with respect to the Boolean operators, specifically the AND and OR operators as they are the most frequently used. It's also necessary to understand this order of precedence for Web Engines that do not offer parenthetical parsing. Fortunately, all of the Web Engines follow the DIALOG lead which has the AND operator taking precedence over the OR operator in a query statement. In the query *cats OR dogs AND animals*, dogs and animals are first ANDed together and the product is then ORed with cats even though the OR operator appears first when reading left to right. The above statement would be equivalent to *animals AND dogs OR cats*. Whether the Web Engines took their cue from the Online Engines or not, it's commendable that they all have standardized the order of precedence for the Boolean operators.

Inclusive vs. Exclusive OR

All of the Web Engines use or claim to use an Inclusive OR as does DIALOG. This means that when searching for *sheep OR cows*, all documents containing either the word sheep, the word cows, or both words will be retrieved. An exclusive OR would not retrieve the documents which contain both words, only the ones which contained one term or the other. Again, we have standardization between all systems which can only help users who use both systems on a regular basis. They do not have to worry about which system uses which type of OR.

Partial term matching vs. Boolean All-or-Nothing

In an AND search, some of the Web Engines will retrieve documents when only one term in a multi-term statement is present in that document. Although they claim so, this is not considered to be a Boolean search. A true Boolean search as performed by the DIALOG Search Engine will return only those documents which contain the specified terms as parsed in the query statement. Nothing more, nothing less. Boolean term matching is exact term matching. Depending on the type of search performed, each type of matching has its benefits. The way certain Web engines perform a search, it is really a combination of an AND and OR search. First the terms are ANDed together to find the best matches which are displayed first in the results listing. Then the documents which have only one (or one less than the total number of terms entered) are listed in the results following the first group of documents where all terms were found. While this is a great strategy for finding all possible relevant documents in a database (a broad search), it often will find many more documents than the user could possibly sort through. To perform a narrower database search, strict Boolean is a much better solution.

Weighted Terms

DIALOG does not use weighted terms when retrieving results. Each and every word entered is of equal value. Either the term is present or it is not. A document is not any more likely to appear in the results list (or appear higher in the ranking) than any other document due to words entered in a search term, although it does retrieve documents in reverse chronological order by date of entry into the database. This is not necessarily true for the Web Engines. They are each a bit different and it is not always apparent how results are retrieved. Their Online help is also of no help in describing how they weight terms. This is discussed more in the Results Section later. For now, it's enough to say that the Web engines do weight words and it does effect the retrieved results. Depending upon the type of search performed, this may or may not be a good feature to use. It works for the Web for novice users who enter simple queries. For the researcher who is accustomed to Boolean logic, a more precise search and retrieval strategy is in order. These two diverse search strategies, while not compatible in one interface, can be said to be of equal value depending on the research being done and the capabilities of the user. Neither is technically an advantage over the other or a more advanced function. They're just differing search methodologies.

Stopwords

Stopwords are important because they stop the user from looking for words that are so common that they are contained in most every record available in the database. They're especially helpful to the relatively inexperienced user who doesn't realize what will be returned when searching for these words. Common stopwords in most databases include, but are not limited to, *and, or, the, a, of,* and *an*. DIALOG, excite, and HotBot all employ stopwords in their engines, even if a listing of those words is hard to find. A message stating that 'the search retrieved no results' displays when one tries to search for a stopword in excite or HotBot. Zero results are retrieved when using DIALOG to search a stopword other than *and* or *or*. When using either of these operators as search terms, the message 'operator "xxx" in invalid position' occurs. AltaVista and Infoseek, however, don't use stopwords. A search for the words *and* or *the* in either engine will return 20 to 30 million hits. When using any of these engines, the novice is likely to be confused as to what is happening and think that the engine is not performing correctly. A better solution would be a short explanation as to why the search didn't perform as expected and a list of all stopwords used by the search engine. Simply stating that the query returned no results without an explanation is not satisfactory. All of the engines could stand some improvement in this area.

Case Sensitivity

Case sensitivity is the distinguishing, by the Search Engine, of upper case from lower case characters. It is important to be able to separate words with the same spelling but different meaning in order to determine relevance or context of a term. For example, the words *windows* and *Windows* are spelled alike but have vastly different meanings. In

order to increase search precision, we would want to be able to do a case sensitive search and reduce the number of false drops in our retrieval. Two of the four Web based engines (Infoseek and AltaVista) employ case sensitive searching. The other two and DIALOG do not. This is certainly a powerful and useful feature that should be contained on all advanced engines. It's certainly an area where the Web Engines are ahead.

META tag vs. Descriptors and Identifiers

The HTML META tag was meant to provide for the input of indexing terms associated with a given Web document. It was planned as a searchable keyword field to be used for simple catalog and keyword searching. The META tag, however, has not gained wide use or acceptance for its intended purpose. It is dependent on the documents author including this field within the HTML document. If it is not, the document can not be searched for using the META field. Because of this lack of support for the META tag, the Web Engines do not include this search option on their interfaces although AltaVista does index this field. Even if every Web page author inserted the META tag, it would still not be as useful as using descriptors and identifiers in DIALOG. The DIALOG databases are indexed and abstracted by professionals who have much experience selecting keywords which describe a given document. Even so, if the META tag were widely adopted it would be an advancement over what is available now which is nothing.

Truncation/Embedded Truncation/Wildcards

The AltaVista, Infoseek, and DIALOG interfaces all provide for both truncation and embedded truncation searching. While using differing wildcard symbols, each allows the user to search for all forms of a rootword or multiple spellings of a word. All three are on par in this category. The HotBot engine does not allow truncation. The excite engine will find all forms of a word if at least three letters are entered. It will not find root words and there is no wildcard symbol.

Phrase Searching

All five of the search interfaces offer some form of phrase searching. All of the Web indexes are both word and phrase indexed. In DIALOG the user must know whether the index being used is word indexed, phrase indexed, or both. Normally, in the DIALOG Basic Index, certain fields are phrase indexed and others are not. One must know this information before searching. It would appear that the Web interfaces are more advanced in this area but we need to remember that it is not practical to phrase index all of the many fields available in a typical DIALOG database. The Web Indexes are only indexing a relatively small number of fields and therefore can limit the size of their indexes. Even so, it would be nice to have all DIALOG fields both word and phrase indexed to create an even more powerful and less confusing interface.

Nested Searching

Again, all five engines offer this feature. Nested searching works on all four Web Engines even though no reference to this feature could be found in the help pages of one of them. All five use parentheses to delineate nested terms and appear to work equally well.

Access Points

It has been said that the measure of good database search software is the number of access points it provides to that database. It is really difficult to make a direct comparison of the number of access points to the databases as the databases are not directly comparable. Suffice to say that both types of interfaces do provide an adequate number of access points to their respective databases. The most advanced options of the Web Interfaces allow access to all available fields except for the META field discussed above. DIALOG normally provides adequate access via dozens of different database access points.

Basic Index

The Basic Index in DIALOG normally consists of the Title, Abstract, Text, Descriptor, and Identifier fields along with other fields depending on the database in use. These make up the most popular and widely accessed fields and are grouped together for ease of use. A free text search in most DIALOG databases will search the Basic Index. The Web interfaces Basic Index equivalent is a search of the entire document itself. This includes the document title, header, text, META tag field, and abstract (where applicable). That is, for those seeking Web documents. Most Web Engines also have the option of searching Newgroups as well. A free text search in the Web databases will not include fields such as URL, host, or link. Surprisingly, both types of interfaces offer essential the same type of Basic Index for searching different types of information.

Multiple Search Sets

DIALOG has the option of creating search sets for each query input to the system. These search sets can then be refined, combined, or subtracted from other search queries or search sets. None of the Web counterparts comes close to the DIALOG functionality in this area. The Excite engine does provide a 'more like this' option to find more pages like the one selected though this isn't the same as being able to use previous searches in the formulation of new ones. The HotBot engine provides a 'revise search' option which lets the searcher revise a search that has already been performed. Still, this does not allow for the combination of sets. And Infoseek will allow the user to narrow the results of the search previously performed by adding terms to that group of results. None of these offer the full functionality of the DIALOG system. Being able to manipulate and refine a search is a basic tenet of Online searching. It is not sufficient for the user to have to re-perform a search from the default interface page every time a query needs to be re-entered. Nor is it acceptable not to be able to intersect two or more independent queries. The Web Engines

are certainly behind and must add this feature to their array of search options in order to be thought of as state-of-the-art engines.

OneSearch Capability

The DIALOG OneSearch capability lets the searcher access more than one DIALOG database at a time using a single interface. Even though the functionality of this interface is greatly limited and one can not form search sets, this function does give the user some indication of how many 'hits' each database contains. The user can then select the databases that contained the most hits for further review. A number of Web sites offer what they call an ALL-IN-ONE search service. These sites are separate from the individual interfaces described in this paper but links from these 'META' Search Engines to the individual engines are provided. Each one of these sites offers a unified interface to multiple search engines. Like OneSearch, these interfaces will simply return a single posting containing the number of hits each Web Engine produced from the search. Links to each of the individual engines are provided for an easy transition to the specific interface selected and are a good way to find the engine most appropriate for a given search. Both OneSearch and the ALL-IN-ONE interfaces provide roughly the same service and can be a valuable time saver.

Natural Language or Concept Searching

Until recently there was no Web Engine that offered Natural Language Searching (NLS). Excite does offer a form of Concept searching with its 'more like this' command. This feature will search for more documents like the one selected from a retrieval listing. But it still isn't a true form of concept searching. It takes a retrieved document, parses it, and finds more documents with like words. A true form of concept searching associates words and differentiates between words spelled the same but with different meanings depending on the context of other words in the document. Infoseek will do this. It does 'Plain English Queries' (NLS) and finds documents containing all variations of a word (mouse/mice). DIALOG doesn't offer a true form of NLS but it does offer a form of rudimentary concept searching with its TARGET command interface. TARGET is really a relevance ranking system. Queries are entered without commands and documents are returned that are considered most relevant by the search software. This type of search can also be thought of as a type of weighted search as certain terms are obviously given precedence over others. Neither system can eally be considered a true Concept or NLS system. It would be nice to have the option to 'turn on' this feature with a simple command or click of the mouse if a search isn't retrieving the desired results using the traditional Boolean operators. NLS may very well be more intuitive for the novice user or one who doesn't particularly want to or have the time to learn all of the parsing rules. Both types of database interface systems could be vastly improved and advanced by offering the full functionalityof these systems in their respective search engines.

Retrieval of Audio, Video, Graphic, and Numeric Files

The command driven version of DIALOG offers only text and numeric files directly. Although some graphics can be ordered via postal mail, there is no provision for transmitting audio, video, or graphics Online. Conversely, the Web tools can search for audio, video, and graphic files by searching for file extensions. Numeric files can not be found in this same way. All this will likely change with the Web based GUI DIALOG has developed called Dialink.

Commands

A comparison and discussion of specific Web Engine and DIALOG commands, or lack thereof, is explored in this section. Whereas the previous section detailed the rules that a query must conform to and the characteristics of the database interface, this section explores the specific commands that all good Search Engines should contain.

Alert Command

The Alert function in DIALOG lets the user save a query to be run when the database is updated. The user does not have to be logged in at the time that the alert profile is run against the database. The alert is automatically run whenever the database is updated with new records. The results can be sent to the user in multiple ways. It is a great way to be kept abreast of new developments in a given discipline or subject area without having to constantly log on to the system and re-enter the same query. None of the Web based engines offers such a function at this time though some specialty Web sites do. One must currently connect to an engine each and every time one wants to find what new information on a given subject or in a given discipline has been added to the engine's database. The engines do, however, offer the 'save search' option discussed later in this section. There are also third party vendors that sell smart agent software which scans the Web for newly created documents that match the user's profile and will then return the URL to the user, normally via e-mail or pop-up icon. The Web interfaces probably have not offered this feature yet due to the large amount of processor space undoubtedly needed to store what would likely become many millions of saved inquiries that would have to be kept track of and run against their databases as often as once a day. If the Web engines ever do become profitable on a continual basis, look for the alert option to one of the first added as any truly competitive engine is needs to offer this function to its very busy users.

Author/Proper Name Command

To find the author or creator of a document, the DIALOG Search Engine utilizes its author field command (AU= or /AU) to search for documents written by a certain individual. From perhaps millions of documents in a given database, the searcher can limit the scope (and retrieval) of the search to only those documents from a particular author. While not providing a directly searchable author field, some of the Web Interfaces do offer the capability of searching for a proper name. This proper name does not necessarily have to be the name of the author of the document and indeed the engine will return documents in which the person whom is being looked for is not the documents author. What it will do is pick out documents where the proper name appears. It is essentially a free text search of the document with one important difference. The names must be capitalized and adjacent to, or close to, one another (a proximity search). Both HotBot and Infoseek offer some form of proper name searching. At this time there is no provision for definitely searching for the document creator. It is a hit or miss proposition. Not all Web documents contain the originators name. Nor are they required to.

Display Sets Command

The Display Sets command in DIALOG is used to display a listing of the previously entered query commands which were formed into search sets. As previously presented, only DIALOG uses these sets to display and combine previous database queries. But what they also do is to inform the seeker what terms were already entered, submitted and how they were combined to form search statements. This can be an invaluable asset to the professional or business user who is not just surfing but is pressed for time and needs to have quick reference to past queries. The only way to display past database queries in any Web Engine is to use the <back> button on the Web Browser. And if one wants to see the first of twenty previous queries, one must click the <back> button twenty times to get back to that first search statement. There is no function which allows one to display all of the input search statements from one session at one time and on one page. It is also not possible to print them out. The only way to view them is to click back and forth to find the desired information.

Expand Command

The Expand command in DIALOG lets the user expand the index of a phrase indexed field. It is used, among other things, to find out how a name or category was indexed, the proper spelling of a proper name, all variations of a name, or the number of times that a particular phrase appears in the database documents. Expanded terms are normally listed alphabetically. There is no comparable command available for any of the Web Search Engines. While one can limit searches to particular fields, one cannot list the contents of those fields to find out how the terms are indexed. The expand command would be potentially most useful if implemented with the META tag. If this were done, it would be possible to expand an index of keywords to find-out how documents are listed. It would almost be equivalent to searching a library subject catalog.

Highlight Command

The DIALOG Highlight command is useful in highlighting those words or terms which were searched for in the query statement in the displayed document itself. The words themselves are not actually highlighted (**bold**) but are enclosed in asterisks (**) so that they will stand out from the rest of the text. This is useful in quickly identifying where the search words appear in the document. Though there is not a directly comparable feature for any Web interface, the equivalent function would be to use the browser's 'find' command to locate a term in the text of a selected document. Also, the term doesn't appear automatically in the retrieval documents. Another step must be taken in order to have the desired term highlighted. Still, none of the leading engines reviewed here offer a comparable highlight command. It would be often useful to be provided with a highlight feature that could be toggled on and off as desired to be able to view the desired search terms in context. The DIALOG highlight command would also do well to provide a bold

function for the highlighted terms when possible. It is much easier to pick out than the asterisks.

Keyword in Context (KWIC) Command

Similar to the Highlight command, the KWIC command in DIALOG will present for display, that portion of the selected document that contains the entered terms. Defaults can be set to display a certain number of words before and after the keywords. Used in conjunction with the Highlight command, passages from the document can be quickly accessed and the context or usage of the keywords easily ascertained. As with the Highlight command, there is no like Web command save for the find function of the browser. The user must first browse the retrieval listing to decide whether to link to the URL. There is no guarantee that the terms entered will appear anywhere in the retrieval listing. The URL must be linked to, and at least a portion of the document read, before the context of the keywords can be determined. For the Web Search Engines, and especially for those that do not distinguish between proper names (capitalization) and common words, having a KWIC option available would certainly aid the time-pressed business researcher.

Limit Command

Each DIALOG database has limiting criteria to screen or filter large portions of the database. Limiting is done by restricting certain fields to certain values. Some of the most popular ways to limit a search are by limiting the retrieval documents to only those written in the English language or from a given publication year. While not being able to limit by language, the Web interfaces do offer a number of limits by which documents can be restricted. For example, Web pages can be limited by country codes. By doing so, we can make a valid assumption that the document has been written in the predominant language of that country. Some Engines offer a limiting by file type while other Engines allow for limiting by source code (i.e. HTML, Java). All of the Web Engines provide a limit by date function of one sort or another. The user must be careful though, because the date used by the Engine is the date that the document was first indexed, not necessarily the date that the information was written or first put on the Web. Both types of Search Tools offer an impressive array of limit functions relative to the type of information contained in their indexes. Eventually, the Web Engines may want to introduce a limit function that distinguishes between professional and nonprofessional sources in an attempt to weed out those many pages that are authored by non-professionals.

Proximity Operators

The common proximity Operators in DIALOG are the W, N, and S operators. The W and N operators are equivalent to, and commonly referred to, as the ADJACENT and NEAR proximity operators. The W operator, placed in-between two terms, will look for those two terms to be next to each other and in the order entered , in a document . The N operator, also placed in-between two search terms, will seek those documents where the

two terms appear next to each and other and in any order. Numbers can be placed before either of these operators to delimit the number of words that the terms must be located within. The S operator will look for the two terms within the same subfield or paragraph. The AltaVista Search Engine is the only Web engine that offers any form of proximity operator searching. It offers the a NEAR command that will find terms within ten words of each other and in any order. There is no provision for changing the number of words that the terms must appear within. It is really a combination of the DIALOG W and N commands only less powerful. This lack of proximity features on the Web interfaces is particularly surprising in that proximity operators are not really considered a highly advanced feature but are considered a basic staple of Boolean searching. In order to truly be on par with their Online cousin, the Web interfaces need to quickly rectify this glaring lack of functionality in their services. They are really missing an easy opportunity to make their Engines much more useful and powerful.

Remove Duplicates Command

The Remove Duplicates feature is a particularly useful feature when searching huge databases containing millions of files or when using the OneSearch function across databases in DIALOG. This command will scan all records in a retrieval listing and determine if there are any duplicate records. It will remove excess copies if any are detected and only present one copy in the retrieval display. Normally a database will not contain duplicative records. This command is used across databases in DIALOG as there is no reason to use it in a single database. It can be valuable in detecting duplicate pages in a single Web Engine index or across multiple Web Engines. The independent Web Interfaces really don't provide a remove duplicates feature. The best that any of them do is list the duplicate pages. HotBot will list all of the URLs that it finds that are linked to the same page. Although it is only one page, it may have a number of different links to it. HotBot will list the page and then each individual link underneath the title. It does not remove the records, it just lists them together to let you know that they're there. The other Web Engines do not explain how they handle duplicate pages. On the Web, it can be either good or bad to not have duplicates removed. It can be good because one link to a site may be down and the user would have the option of connecting to the desired information via another (mirror site) link. But it could also be bad because a site may have literally hundreds of links to it. One would not want the first 100 documents in a retrieval listing to be for the same document. What is needed is a toggle option to either remove duplicates or leave them in place depending on the user's preference for a given search.

Save Search Command

The only Web Engine of the four to offer any form of Save Search command is the HotBot Search Engine's 'Save My Settings' option. Theoretically it will save the user defined settings from a search to run at a later date. These settings would include both the terms entered as well as other selected features (i.e. file type, date). Unfortunately, the atuthor has not been able to get this feature to work. DIALOG does provide a functioning Save Search option thankfully. The user simply enters the save command and

gives the saved search a name. Later, in the same session or another session, the user can run the search in the same database or another DIALOG database. If the Web Engines did ever get this feature properly installed, it could serve as a stop-gap alert feature as well. Assuming 'free' or unlimited Web access, it wouldn't take much time to run these saved searches on occasion to check for new Web page additions to the index. Though memory consuming like the Alert functions, this is really a feature in need of a home on the Web. Even if the 'free' Web Engines had to charge a nominal fee for the service, it should still be added to their list of interface options.

Sort Command

The Sort command in DIALOG will let the user sort the retrieval documents (hits) before a listing of titles or abstracts is displayed. Depending on the database, the results can be sorted and displayed by any number of combinations of fields allowed by the database being searched. Web documents are not normally sorted in any particular order. The only way to order a set of Web documents using the Web interfaces, is to enter keywords in the 'must contain' query box on the interface page. This will list those documents that contain the entered keywords first in the retrieval listing. Also, as explained previously, the Web engines will weight certain words and return documents containing those words higher in the retrieval listing. Obviously, this is a poor man's substitution for a true sort function. The ranking function for Web documents is not the same as the Sort command in DIALOG and will not provide definitive results. As they are set up today, no Web Engine can sort a listing of hits by field for the user.

Stack Commands

The stacking function allows the user to enter multiple commands at once. Because the Web Engines do not offer the full range of search options that the DIALOG Engine does, entering multiple commands isn't as intuitive or logical with the Web Engines. With the DIALOG Interface, the user can enter a search statement and print out the first record using the same interface stacking commands. The best Engine that comes closest to this stacked command option is the HotBot Search Engine. From a single interface screen, a researcher can enter a search statement, restrict the results by date, restrict the results further by file type or country, and select a method of displaying the results. Though not a particularly imperative feature to have, once again we see a time saving feature that has not yet been implemented fully on any Web Engine search page.

Output

A general comparison of retrieval output and record display options and features is discussed in this section.

Output and Results Display Choices

This section introduces and evaluates the differing default search result outputs generated by the different engines and the options for changing the output format. The standard output default for the Web engines is to display the page title, URL, a summary or abstract, and a relevance ranking expressed as either a % confidence ranking or in a numbered fashion with those documents which 'best' fit the query appearing towards the top of the list. Some will also display document size and/or the date that the document was last modified or last indexed by a given spider. Certain engines will tell the user how, and by what criteria, pages are ranked, while others will simply refer to an unexplained algorithm that decides how pages are ranked. HotBot will tell you that it ranks pages by word frequency, words in titles, META tag words, and the ratio of search words to document length. Perplexingly, AltaVista's advanced search is the only one of the Web engines to not rank output. According to the AltaVista search page, those results are ranked "in no particular order".

All of the Web engines will let the user modify the output display to some extent. In addition to the standard output format described previously, AltaVista and HotBot allow the user a compact (brief) and detailed (full) output option. The former will also output a simple 'count only' display while the latter will allow a URL only display. Excite and Infoseek contain somewhat different options in addition to the default or standard option. Excite offers a 'view titles only' and a 'view by website' option. The first option will give a simple one line title listing, while the second lists links grouped by the 20 websites with the most links retrieved. Infoseek's only discernible option other than the default is a 'hide summaries' option which simply does such and condenses the display. All output options described above provide links to the listed pages and each returns 10 documents per results page using the default option settings.

Until recently, none of the Web Engines permitted the reuse of search output. Only lately has AltaVista introduced their 'LiveTopics' option which permits the user to refine a search by selecting from a number of predefined terms. Basically, the user is simply ANDing terms to the result set to achieve a smaller retrieval set. Infoseek, even more recently, has started a 'search only these results' option on its output screen. This also is basically an ANDing feature.

The DIALOG search results are not returned in a standard output format. The user selects from a number of formats including KWIC, Bibliographic Citation, or Full text of the document. Specialized options are also available to display individual fields only. Unlike the Web engines, DIALOG provides no ranking of the documents returned. It simply lumps them all together, giving each an equal weight. Documents are normally

returned by having the most recent record added to the database placed at the top of the hit list (reverse chronological order). By no means, however, does this mean that the most recently added documents are the most recently updated and contain the most up-to-date information. By using the special field options, though, it is possible to perform certain amount of document ranking using the SORT and RANK commands though these options may not be available in all databases and with all types of data. File size in bytes is not an option. However, the user can specify that the document's word count be included in the output. No documents are actually returned until the user decides to display a set of records. Only the number of records available for that search is shown. Once the documents are displayed, they are shown in such a way that there is a continuous scroll through the documents. In other words, all of the documents are displayed on a single page. DIALOG undoubtedly excels in providing the greatest and most flexible number of output formats. But where DIALOG really becomes a much more valuable search tool is in the manipulation of output. Whereas the Web interfaces allow only ANDing of output to narrow a search, DIALOG allows full Boolean searching in refining output. In fact, all of the options used in the initial search are again available in all subsequent searches. The searcher need only create initial search sets that can be manipulated in unending ways later on.

Advanced Features

This section briefly looks at a few of the advanced output features either included in, or lacking from, the engines.

One problem with HTML documents in a retrieval list is that each one must be accessed individually to either view or print a document's contents. There is no way to print documents number 1,5, and 10 from a listing without hyperlinking to a document. While HTML hyperlinking of documents can be a valuable tool, this is one instance where plain text documents facilitate speedy document retrieval. However, with text documents in DIALOG, there is one added step in printing or viewing a document. The user must first display a titles list of documents and then enter a type, display, or print command. Depending on the specific use that the user intends, one method may or may not have an advantage over the other.

Another advanced feature consideration worth looking into is document delivery. Both systems offer functional options for receiving documents. Web documents can be e-mailed to the user's account or another's e-mail account, though the document will appear with all HTML tags and formatting. Another option is to cut and past the document from the Web browser to a word processor. Or the user can simply send the Web page to a printer. Any of these options provide instant document delivery and therefore access to the document information. DIALOG provides a document capture function when telneting as well as e-mail delivery of most documents and alert profiles. Although DIALOG provides a mail service for professional quality documents, in today's fast-paced business climate, realistically, the end-user cannot wait even a day or two to receive the

document. Perhaps DIALOG's Document Delivery will improve with the introduction of its GUI on the Web.

Sample Searches

The following search vocabulary or query samples are used as examples to find clues as to how the best of the Web Engines and the DIALOG engine handle special search cases. We did not evaluate response time, perceived database value, or comparison of documents retrieved. We were only interested in how each system handled special characters or search characteristics. DIALOG files 709 (Akron Beacon Journal) and 725 (Cleveland Plain dealer) were used for all searches as it was felt that there would be ample references to all search terms used in this section. The actual searches confirmed numerous references to each term.

False Drop/Word Context Search

This first search was performed in order to determine if any of the systems could distinguish between two differing contexts of a word and had features or functions available to do so. The sample words input to the systems for this exercise were the terms *Windows/windows* and *Turkey/turkey*. Neither the Web engines nor the DIALOG engine include context capabilities or intelligent agents to perform these searches. Though some of the Web engines are case sensitive and one can narrow a search greatly by simply using the proper form of the word, there were still far too many instances where documents containing references to *turkey*, the bird, were returned when the term *Turkey* (capitalized) was input. HotBot, Excite, and DIALOG all returned an identical number of documents when either form of the term was entered. While AltaVista allows the user to narrow a search using its LiveTopics feature, it still presents the user with the options of adding (ANDing) the term *salad* to the results when *Turkey* was input and gave the option of adding *Syria* as a term when the *turkey* form of the word was input. It could not distinguish the different forms of *windows* either. Excite also allows the user to refine a search by adding terms, though it suggested both *Turkish* and *thanksgiving* as terms to add when either form of the word was searched on. As for *windows*, both searches returned only computer terms as suggestions. It treated the word *windows* as a computer term exclusively. The same problems were encountered when using Infoseek as when using Excite. A partial solution to this problem is to use certain fields in DIALOG such as product, event, or SIC codes in refining a search. Another is to use the KWIC feature to display excerpted document passages in order to determine the use of a word. Of course, this can be extremely time consuming and not worth the effort. Still, none of these engines is able to determine the context of a word. Intelligent agents and natural language searches need to be employed.

Acronym Search

This search was performed in order to determine how white space and/or periods between letters in acronyms are dealt with and whether there was an identifiable way to search the different forms of the acronym. The acronym NASA (N.A.S.A.) which stands for the *National Aeronautics and Space Administration* was used in all searches.

All five systems had no problem finding references when inputting either *NASA* or *nasa*. Infoseek, Excite, and AltaVista were the only search systems of the five to return hits for *N.A.S.A.* with Excite ignoring the periods and AltaVista treating them as white space. The other two gave error messages or zero returns. AltaVista was the only one of the five to correctly search for *N A S A* (spaces). The others returned error messages or zero hits except for Infoseek which apparently looked for the individual letters. In searching the DIALOG database using proximity operators, searches for *N()A()S()A and N(w)A(w)S(w)A* retrieved zero hits. Because the only way to search the DIALOG database for acronyms is by using the actual string of letters with no white space or punctuation, it is obvious that Web engines do a much better job of providing for all possible forms of an acronym.

Common Stopwords

A search of commonly occurring words was performed to reveal which words an engine would not search for, if any.

Stopwords are commonly occurring words in the English language that are likely to be present in most documents. Examples include, but are not limited to, *the, a, an, and, to, of, if, or* and *in*. Some stopwords also act as Boolean operators. HotBot, Excite, and DIALOG all either ignore common stopwords such as those listed above or return a syntax error message when the stopwords are entered individually as terms as would be expected of all quality search software. Infoseek performs an actual search on the term and returns millions of documents. This can be extremely confusing to the novice user and really needs to be corrected by Infoseek. AltaVista does something strange. For its simple search mode, terms are ignored as would be expected. Inexplicably, with its advanced search, AltaVista performs the search and returns millions of documents as Infoseek did. A likely explanation is that due to the powerful search capabilities of its advanced search interface, having a stopword list was sacrificed in favor of optimal search power and options. As novice Web searchers are not as likely to use the advanced query option as are more proficient users, not having stopwords is an understandable characteristic of this engine.

Hyphenated Search

Similar to the Acronym Search, a Hyphenated Search was performed on all databases to see how well commonly hyphenated words could be retrieved and how each system treated such special cases. A search engine's ability to distinguish between words that appear in the same document and words that actually go together is very important in narrowing a search. For example, searches were performed using the terms *Parke-Davis* and *Parke AND Davis*. The latter resulted in over 1000 times as many documents being returned by the engines on average as did the former. Obviously, it's very important ans valuable for search software to be able to search for hyphenated terms.

All five engines handled hyphenation differently. DIALOG has the best solution. The only engine that can not handle hyphens at all is Excite. The hyphenated terms *Park-Davis* and *e-mail* were used as representative hyphenated terms.

As stated, Excite will not perform a hyphenated search. One must use a space instead of a hyphen to get any results. In Infoseek and AltaVista, both *Parke-Davis* and *"Parke Davis"* searched as a phrase returns an equal number of hits. And in HotBot, hyphens are ignored and treated as whitespace. The only engine which adequately compensates for hyphenated words is DIALOG. By using the *(1w)* option in place of the hyphen, the user is able to sufficiently find all forms of a term, whether hyphenated or not. Searching for *Parke()Davis* found all instances where the terms *Parke* and *Davis* occurred within one word of each other and in that order. So, no matter if the term is hyphenated by an author or not, the term is found by the DIALOG search software. None of the Web engines can do this.

Same Word Search

A reliable search interface should be able to distinguish when the exact form of a word is entered twice and notify the user. A reliable engine should not return a results listing of documents. An error message should display. Not one of the engines returned a note that the search used incorrect syntax. The statements *baseball AND baseball* and *baseball OR baseball* were entered into each of the interfaces. In all cases the engines returned an equivalent number of hits when the two searches were performed in each given engine. Even if the engines were looking for multiple occurrences of the term in a given document, certainly the ORed search would return a greater number of hits than an ANDed search using identical terms. While not a glaring defect, again, the novice user is not apt to comprehend the significance of the results of a search should that person perform such a search. All the engines need to improve this lack of notification to the user that this statement is an invalid statement.

Root Word Search

This final search was performed to determine whether the engines strip terms to the rootword or find only the exact word. The terms *surf, surfer,* and *surfing* were input into each search engine to see if the number of hits returned was equivalent for each or the variations of the term. If so, then the engine searches for the root form of the term by default. If not, then it treats each term as a separate entity. The preferred form of searching would be for the search software to treat each word as unique but also provide a truncation feature to search for all forms or variations of the word. All of the search tools performed well using the preferred method. They each returned a different number of hits for each of the three terms. This is how an experienced search would expect the software to behave and is certainly the most intuitive form of performing this type of searching.

Analysis of Search Engine Features

A comparison of the most advanced features of the Web search interfaces to the DIALOG search interface has been summarized by highlighting those components of each given system that were found to be lacking and/or most advanced. Recommendations for improvement are provided.

There are a minimum set of required options that can be said to be expected of any search engine software. These would include a mechanism for searching on and combining terms, a robust search language, and a method of displaying and outputting of retrieved documents. While all of the engines analyzed did meet these bare minimum standards, to be a player in the competitive search software marked, any engine worth using really needs to have a much greater array of search features and options available to the user.

The clear leader of the search engine software evaluated in this paper is undoubtedly the DIALOG search software. While not perfect nor as robust as it could be, DIALOG clearly markets to a more professionally trained user community which is used to database searching and using advanced search techniques on a regular basis and thus expects to have those advanced functions available.

The few areas where DIALOG does falter are as follows. It can be argued that a basic minimum requirement of a good search engine would be the ability to distinguish case sensitivity. As the sample search in this paper clearly demonstrates, two completely different contexts of a word that is spelled the same will yield vastly different documents. Retrieving a set of documents that contains 50% garbage, does not a good search make. Another area found obviously lacking with this engine is its ability to do Natural Language Searching (NLS) or term weighting. A suggested improvement would be to have a SET WEIGHTING ON option to allow the user to specify those terms most important and thus the ranking of documents in a large retrieval listing. Other less prominent areas that DIALOG would do well to offer additional functionality include the areas of Document delivery and the transfer of video, audio, and graphics files. These will likely be addressed by their new Web GUI.

The Web search engine interfaces, while far more advanced than when they first appeared on the Web, still need to have a greater robustness, functionality, and number of database access points in order to be thought of as up to par with the professional (fee) vendors. This was what was and is to be expected when providing a service for 'free'. After all, you get what you pay for.

The greatest advantages that the DIALOG engine has over its Web counterparts is its ability to refine search results, create multiple search sets for later use, and in its use of Descriptors, Identifiers, and Event Codes to classify documents for precise retrieval. The ability to manipulate search subsets and to refine a search to narrow or expand a hit list is a fundamental database query necessity. Being able to perform only one search at a time

severely restricts the user's ability for find useful, as well as pertinent, information. Having sôme sort of classification scheme in place and having professional indexers assign terms to database documents is an invaluable asset for information seekers. In essence, when using this option, a good deal of the searchers work is already performed and that user does not need to wade through useless documents to find possible hits. While the employment of the META tag is a good start in this direction, there isn't an easy solution to this problem on the horizon due to the nature of the data itself. Nonetheless, the Web engines do need to start actively encouraging users to search the META field.

Other needed improvements in the Web interfaces include the ability to save a search query, the ability to receive alerts when the database is updated, and the ability to view a search history on a single Web page. Again, these features should be standard fare on all database interfaces in use today. Anything less, and the engine can't be though of as a top tier search engine. Other, less important features that would certainly add additional functionality to their engines, would be a sort command, a remove duplicates command, a highlight/KWIC command, an increased ability to perform proximity searching, and an expand command (used in conjunction with the META tag perhaps). A few final suggestions to the Web engine designers. Provide a list of non-searchable stopwords for novice users. Expand your Help pages to include examples for all available features your engine offers. More clearly define why it is that certain documents appear higher in a retrieval listing than others. And provide a full listing of all your features, functions, and options on one page for ease of use.

V Summary and Conclusions

This paper was intended to bring to the fore a number of issues related to Search Engine options, features, and characteristics that one should consider when performing search queries using the respective services. It was intended to be a comprehensive comparative analysis of the popular Web Search Engine search interface features to the established (fee) Online database vendor search interface features. It was also intended to present suggestions for improvement to those areas of the Web interfaces found lacking. I hope it has done an adequate job.

The World Wide Web (WWW) public access Search Engines, AltaVista, Excite, HotBot, and Infoseek, still do not contain a number of the advanced commands, options, and features commonly available with the for-profit Online database user interfaces. It was found that the Web search interfaces, as a whole, still trail the DIALOG search interface in terms of the quality, quantity, depth (robustness), and usability of the search system.

While this paper has generally sided with adding more and better features to the search interfaces, it is recognized that with advanced feature implementation, there is a risk of getting too far ahead of the average user's learning curve. Not all users are professionally trained searchers and there may still be a market, and indeed a preference by some, for using simplistic query interfaces.

It is evident, by viewing the short history of Web Search Engine development, that the engine interfaces are in constant change and undergoing continuous development. It is assumed that the issues presented here will need to be revisited on a regular basis for the foreseeable future.

VI Appendixes

Appendix A

Web Engines

Background Information

	AltaVista	excite	HotBot	InfoSeek Guide
Owner	Digital Equipment Corporation	Excite, Inc.	Hotwired, Inc.	Infoseek, Inc.
Date Started	December 1995	December 1995	May 1996	Early 1995
Cost	Free	Free	Free	Free
URL's Indexed	30 million +	50 million +	50 million +	30-50 million +
How Pages Added	Robots/Submittal	Submittal	Submittal	Robots/Submittal
Documents Reviewed?	No	No	No	Yes
Scope	Web/Usenet	Web/Usenet/Excite Reviews	Web/Usenet	Web/Usenet/ Infoseek Select Sites/Companies/ Web FAQs
Subject Catalog/ Directory?	No	Yes	No	Yes
Catalog Updated How Often?	Monthly	Weekly	Monthly	Monthly
Index Update Date?	By URL	None	By URL	By URL
Advertising?	Yes	Yes	Yes	Yes
Sell Words?	No	Yes	Yes	Yes
Online Help?	Yes	Yes	Yes	Yes
Web Site Reviews?	No	Yes	No	Yes
Customize Start Page?	No	Yes	No	Yes
Support META tag?	Yes	No	Yes	Yes
Robot Search Level	3+	3+	Unlimited	1
URL Check	Yes	No	No	No
Spam Penalty?	Yes	Yes	Yes	Yes

Search Parameters

	AltaVista (Advanced)	excite	HotBot	InfoSeek
Fields Indexed	Every Word	Every Word	Every Word	Every Word
Fields Available for Searching	Host/Link/Title/ URL/Language /Date/Domain	None	Date/Domain Geographic Area MediaType/Extens ion/Page/Title	None
Default Search	Phrase	OR (Enclusive)	OR	OR (Exclusive)
Boolean Operators Used	AND/OR/AND NOT	AND/OR/AND NOT	AND/OR/NOT	AND/OR/NOT
Use + or - ?	Yes	Yes	Yes	Yes
Nested Boolean ()	Yes	Yes	Yes	Yes
Proximity Operators Used	NEAR (terms w/in 10 words)	None	None	NEAR (terms w/in 100 words) ADJACENT
Truncation	Yes - use after at least 3 char. Limit of 5 char's. *	Yes - automatic	No	Yes
Wildcards	Yes *	No	No	No
Phrase Search	Yes ""	Yes ""	Yes "" or use the 'the exact phrase' option	Yes ""
Case Sensitive?	Yes - When CAPS used in query	No	No	Yes
Stop Words	None	a, an, and, are, be, if, in, is, it, of, on, the, then, to, when, while, why...	a, an, and, are, be, if, in, is, it, of, on, the, then, to, when...	None
Search for Numbers/Symbols ?	Yes/No	Yes/No	Yes/No	Yes/No
Support META Tag?	?	No	?	Yes
Concept (NLS) Searching?	No	Not Very Good	No	Not Very Good
Form Multiple Search Sets?	No	No	No	No
Refine Search results?	Yes	No	Yes	No
Weights Keywords?	Yes	Yes	Yes	No
Domain Search?	Yes	No	Yes	No
Search by Date?	Yes	No	Yes	No
Query by Example?	No	Yes	No	Yes
Sets - Combine Multiple Search?	No	No	No	No
Operator Order Precedence	AND, OR, NOT	AND, OR, NOT	AND, OR, NOT	AND, OR, NOT

Output

	AltaVista	excite	HotBot	InfoSeek
Ranking System	No - unless choose important terms	Yes	Yes	Yes
Duplicate Detection?	Yes	Yes	Yes	Yes
Documents Per Page/Total	10/Unlimited	10	10	10/100
Display Options?	Yes	Yes	Yes	Yes
Save Results as Bookmark?	Yes	No	?	Yes
Set Max. Number of Hits Returned?	No	No	No	No
Set Number of Hits Displayed?	Yes	No	Yes	Yes
Site Popularity	No	Some	No	No
Size of Document Displayed?	Yes	No	Yes	Yes
Refine Results?	Yes	Yes	Yes	Yes
Site Reviews?	No	Yes	No	Yes

Appendix B

DIALOG

Background Information

	Dialog
Owner	Knight-Ridder Information
Date Started	1960's
Cost	Varies
URL's Indexed	N/A
How Pages Added	Vendor/Publisher
Documents Reviewed?	Yes - Various Levels of Review
Scope	Varies
Subject Catalog/ Directory?	Yes
Catalog Updated How Often?	Varies - Daily/Weekly/ Monthly
Index Update Date?	Varies - Daily/Weekly/ Monthly
Advertising?	No
Sell Words?	No
Online Help?	Yes
Web Site Reviews?	No
Customize Start Page?	No
Support META tag?	N/A
Robot Search Level	N/A
URL Check	N/A
Spam Penalty?	N/A

Search Parameters

	Dialog
Fields Indexed	Word/Phrase
Fields Available for Searching	Varies w/ Database
Default Search	Word/Phrase in Basic Index
Boolean Operators Used	AND/OR/NOT
Use + or - ?	No
Nested Boolean ()	Yes
Proximity Operators Used	NEAR/ADJ/W Variable
Truncation	Yes
Wildcards	Yes ?
Phrase Search	Yes
Case Sensitive?	No
Stop Words	an, and, by, for, from, of, the, to, with
Search for Numbers/Symbols ?	?/No
Support META Tag?	N/A
Concept (NLS) Searching?	TARGET
Form Multiple Search Sets?	Yes
Refine Search results?	Yes
Weights Keywords?	No
Domain Search?	N/A
Search by Date?	Yes
Query by Example?	No
Combine Multiple Search Sets?	Yes
Operator Precedence Order	AND, OR, NOT

Output

	Dialog
Ranking System	No - Rertrieved by Date Entered in Database
Duplicate Detection?	Yes
Documents Per Page/Total	Unlimited
Display Options?	Yes
Save Results as Bookmark?	N/A
Set Max. Number of Hits Returned?	No
Set Number of Hits Displayed?	Yes
Site Popularity	N/A
Size of Document Displayed?	No
Refine Results?	Yes
Site Reviews?	N/A

VII Bibliography

(1) Balas, J., Online Treasures: Staying Afloat in the Net Sea. Computers in Libraries, May 1997. P. 41-43.

(2) Barlow, L., The Spider's Apprentice: How to use Web Search Engines. November 1996. [http://www.monash.com/spidap.html]

(3) Barry, T. and Richardson, J., Indexing the Net: A Review of Indexing Tools. AUSWeb96: Landscaping the Web, Gold Coast, Queensland, Australia, July 7-9, 1996. [http://www.scu.edu.au/ausweb96/educn/barry1/paper.html]

(4) Bocher, B., A higher signal-to-noise ratio: Effective use of Web Search Engines. WETC: Wisconsin Educational Technology Conference, Green Bay, Wisconsin. October 9, 1996. [http://www.state.wi.us/agencies/dpi/www/search.html]

(5) Brandt, D., Relevancy and Searching the Internet. Computers in Libraries, Volume 16, Number 8, September 1996. P. 35, 38-39.

(6) Carr, D., New Search Engine Targets Intranet Users. WebWeek, Volume 2, Issue 20, December 16, 1996. P. 48

(7) Chu, H., and Rosenthal, M., Search Engines for the World Wide Web: A Comparative Study and Evaluation Methodology. ASIS '96 Proceedings of the 59th Annual Meeting, Baltimore, Maryland, October 21-24, 1996. Volume 33. P. 127-135.

(8) Cobb Group, Choosing the right Internet Search Tool. Inside Netscape Navigator, May 1997. P. 5-9.

(9) Conte Jr., R. Guiding Lights. Internet World May 1996. P. 40-44 [http://www.iw.com/1996/05/guiding.html]

(10) Courtois, M., Cool Tools for Web Searching. An Update. Online, May/June 1996. Volume 20, Number 3, P. 29-36.

(11) Ding, W., and Marchionini, G., A Comparative Study of Web Search Service Performance. ASIS '96 Proceedings of the 59th Annual Meeting, Baltimore, Maryland, October 21-24, 1996. Volume 33, P. 136-142.

(12) Dong and Su, Online & CDROM Review, April 1997. Volume 21, Number 2.

(13) Duncan, G., Send In the Robot. Macworld January 1997. P. 152-156.

(14) Feldman, S. Testing Natural Language: Comparing DIALOG, TARGET, and DR-LINK. Online, November/December 1996. P. 71-79.

(15) Fleishman, G. Searching the Surf (Part 1 of 2). WebSpy Issue #14 October 15, 1996. P. 1-10 [http://www.adobemag.com/Departments/webspy.html]

(16) Garman, N. Be a Savvy Online Consumer. Online User July/Aug 1996. P. 5.

(17) Getting Started on DIALOG: A Guide to Searching. DIALOG Information Services, Inc., 1994.

(18) Gray, T., How to Search the Web: A Guide to Search Tools. July, 1996. [http://issfw.palomar.edu/Library/TGSEARCH.HTM]

(19) Hearst, M., Special Report: Interfaces for Searching the Web. Scientific American. March 1997. [http://www.sciam.com/0397issue/0387hearst.html]

(20) How Search Engines (Say They) Work. July 24, 1996. [http://maxonline.com/webmasters/work.htm]

(21) James-Catalano, C.N., One-Stop Surfing. Internet World November 1996. P. 32-34.

(22) Koch, T., Searching the Web: Systematic Overview Over Indexes. October, 1996. [http://www.ub2.lu.se/tk/websearch_systemat.html]

(23) Lager, M., Spinning a Web Search. Untangling the Web: Proceedings, University Center, University of California, Santa Barbara, April 26, 1996. [http://www.library.ucsb.edu/untangle/lager.html]

(24) Liu, J. Understanding WWW Search Tools, Second Update Reference Department, IUB Libraries. September 1996. [http://www.indiana.edu/~librcsd/search/]

(25) Lynch, C. Special Report: Searching the Internet. Scientific American. March 1997. [http://www.sciam.com/0397issue/0397lynch.html]

(26) Morgan, C. The Search is On. Windows Magazine November 1996. P. 212-230.

(27) Northwestern University Library, Evaluation of Selected Internet Search Tools. July 26, 1997. [http://www.library.nwu.edu/resources/internet/search/evaluate.html]

(28) Notess, G., Internet "Onesearch" with the META Search Engines. Online, November 1996. [http://www.onlineinc.com/onlinemag/NovOL/nets11.html]

(29) Overton, R., Search Engines get Faster and Faster, but not always better. PCWorld, September 1996. [http://www.pcworld.com/workstyles/online/articles/sep96/1409_engine.html]

(30) Page, A. The Search is Over: The search-engine secrets of the pros. PC Computing October 1996. P. 143-154.

(31) Pfaffenberger, B. Web Search Strategies. 1996.

(32) Singh, A., and Lidsky, D., All-Out Search. PC Magazine, December 3, 1996. P. 213-249.

(33) Slot, M., The Matrix of Internet Search Catalogs and Engines. June, 1996. [http://www.sils.umich.edu/~fprefect/matrix/matrix.shtml]

(34) Solock, J. Searching the Internet Part II: Subject Catalogs, Annotated Directories, and Subject Guides. 1996. [http://rs.internic.net/nic-support/nicnews/endusers.html]

(35) Srinivasan, P., Ruiz, M., and Lam, W., An Investigation of Indexing on the WWW. ASIS '96 Proceedings of the 59th Annual Meeting, Baltimore, Maryland, October 21-24, 1996. Volume 33, P. 79-83.

(36) Steinberg, S., Seek and Ye shall Find (Maybe). Wired, Volume 4, Number 5, May 1996. [http://www.hotwired.com/wired/4.05/features/indexweb.html]

(37) Stellin, S., Can You Trust Your Search Engine? CNET Features 1996. [http://www.cnet.com/Content/Features/Dlife/Search/]

(38) Strategic Alliances and Victories. July 24, 1996. [http://maxonline.com/webmasters/compare.htm]

(39) Sullivan, D. and Karpinski, R., Supercharge your Web Searches. NetGuide. May 1997. P. 63-84.

(40) SunSITE, Internet Search Tool Details, September 1996. [http://sunsite.berkeley.edu/Help/searchdetails.html]

(41) Tillman, H. Evaluating Quality on the Net. July 10, 1996. [http://www.tiac.net/users/hope/findqual.html]

(42) Tomaiuolo, N. G., and Packer, J. G., Quantitative Analysis of Five WWW Search Engines: Results of 200 Subject Searches. Computers in Libraries Volume 16, Number 6, June 1996. P. 58-62.

(43) Venditto, G. Search Engine Showdown. Internet World May 1996. P. 79-86.

(44) Web Matrix: What's the Difference? October 30, 1996. [http://www.sils.umich.edu/~fprefect/matrix/answers.html]

(45) Westera, G., Robot-Driven Search Engine Evaluation Overview. October 1996. [http://alpha6.curtin.EDU.AU:80/curtin/library/staffpages/gwpersonal/senginestudy/index.htm]

(46) Zorn, P., Emanoil, M, Marshall, L., and Panek, M. Advanced Web Searching: Tricks of the Trade. Online Vol. 20 No. 3, May/June 1996. P. 15-28.